Report on Semantic Segmentation and Analysis of Videos

- By Yash Bhaskar

This report details the implementation of a semantic segmentation pipeline applied to YouTube videos. The goal is to extract meaningful audio-text pairs from the video, ensuring each pair is semantically coherent and under 15 seconds in length.

Workflow and Implementation:

1. Download Video and Extract Audio:

We utilized the pytube library to download the video and moviepy to extract the audio.

2. Audio Preprocessing:

To enhance the transcription quality and mitigate potential issues with noise, we applied the following audio preprocessing steps:

Noise Reduction: We used the noisereduce library to reduce background noise. noisereduce is a Python package that implements various noise reduction algorithms, specifically designed for audio signals. This helps reduce "hallucinations" in the transcription process.

Audio Resampling: We resampled the audio to 16,000 Hz using librosa. librosa is a powerful Python library for audio analysis and manipulation, including resampling. This ensures compatibility with most speech-to-text models like Whisper and wav2vec2, which use 16,000 Hz as their default sampling rate.

3. Transcription of Audio:

We experimented with two open-source speech-to-text models for transcribing the audio:

1. **Facebook wav2vec2 Model:** This model is known for its speed and efficiency. However, it struggles with transcribing large audio files (longer than a few minutes). This is because it computes the entire transcript at once, making it computationally challenging for extended audio segments.

   Analysis: To address this limitation, we initially considered using Voice Activity Detection (VAD) to split the audio into chunks and feed them individually to the wav2vec2 model. However, we hypothesized that splitting the audio based on non-voiced regions might lead to inconsistent results due to the model's training on sentence-level data. Additionally, managing overlaps and missed audio segments would require significant preprocessing and postprocessing, adding complexity to the workflow. Therefore, we decided to move forward with the Whisper model.

2. **Open AI Whisper Model:** This model is renowned for its accuracy and robustness across various audio formats and languages.

   Analysis: During our testing, we observed that Whisper occasionally exhibited "hallucinations," repeating the same text segment multiple times, especially when encountering highly noisy audio portions or when noise reduction algorithms unintentionally removed crucial features.

4. Time-Align Transcript with Audio:

Whisper inherently provides timestamps for each word or phrase in the transcript. This allows for a direct and accurate alignment between the text and corresponding audio segments. No additional alignment tools were necessary in this case.

5. Semantic Chunking of Data:

The audio is segmented based on the timestamps provided by Whisper for each transcribed segment.
Each segment represents a distinct word or phrase, ensuring semantic coherence within each chunk.
This approach automatically identifies natural sentence or phrase boundaries, eliminating the need for additional semantic analysis techniques.

Strengths and Weaknesses:

Strengths:

1. Whisper provides highly accurate transcription, timestamping in chunks in a single pipeline making the process easier and faster.
2. The automatic semantic chunking based on Whisper's output effectively identifies natural boundaries, resulting in meaningful segments without the need for identifying semantically rich segments **but it has some drawbacks discussed below.**

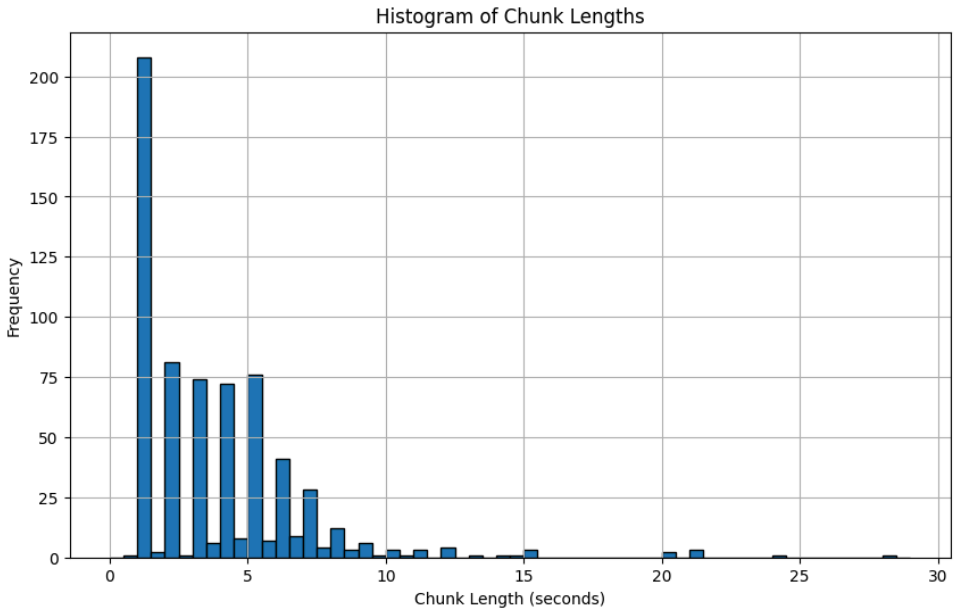
Weaknesses:

1. Whisper's transcription sometimes outputs the same sentence text again for a small time span.


```
{
    "chunk_id": 439,
    "chunk_length": 1.0,
    "text": " I am going to make a model which is not a model.",
    "start_time": 1559.0,
    "end_time": 1560.0
},
{
    "chunk_id": 440,
    "chunk_length": 1.0,
    "text": " I am going to make a model which is not a model.",
    "start_time": 1560.0,
    "end_time": 1561.0
},
{
    "chunk_id": 441,
    "chunk_length": 1.0,
    "text": " I am going to make a model which is not a model.",
    "start_time": 1561.0,
    "end_time": 1562.0
},
{
    "chunk_id": 442,
    "chunk_length": 1.0,
    "text": " I am going to make a model which is not a model.",
    "start_time": 1562.0,
    "end_time": 1563.0
},
```


As we can see here, we know that the same sentence must not have been said 4 times 1 sec each. But some post processing can fix this issue.

2. Below is the graph for Chunk Length (in seconds) vs Frequency Histogram:



Here as we can see the frequency for chunks near 1 sec is very high and also its very unlikely that
chunks of 1 sec will have rich semantic information. So although Whisper segments the transcripts, its not perfect.

To fix it, we need to combine some chunks which can be decided by some model trained on
taking a chunk / phrase and give a score based on its semantic richness and completeness. OR we
can just prompt an LLMs (or make an Agent for it) to determine if 2 chunks should be combined
or not.

3. Whisper does not support transcription in multiple languages. Eg if the audio has multiple languages say English and Hindi, then the translation will only be in the language that is specified in the pipeline, or if not specified then the language spoken in the start of the audio is used

Conclusion:
The implemented semantic segmentation pipeline successfully extracted meaningful audio-text pairs from the video. The approach leverages the capabilities of Whisper for both transcription and timestamping, simplifying the process and achieving high accuracy. The inherent semantic boundaries identified by Whisper ensure the coherence and meaningfulness of each segment. While the approach exhibits certain limitations, its overall performance and generalizability make it suitable for a wide range of video analysis tasks.

Appendix :

Gradio Setup: